

Nama : Beautifully Margaretha P. W

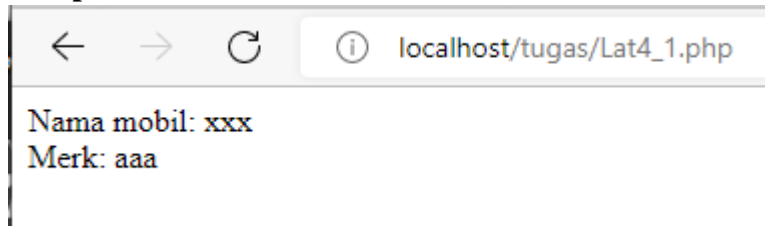
NIM : G.211.21.0081

Matkul: Pemrograman Framework

Project Praktek OOP pada PHP

- **Latihan 1: Class dan Object + Modifier**

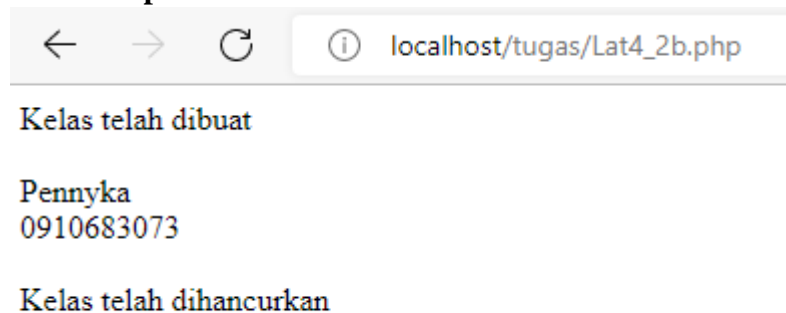
- a. Tampilan dari source code:



```
localhost/tugas/Lat4_1.php  
Nama mobil: xxx  
Merk: aaa
```

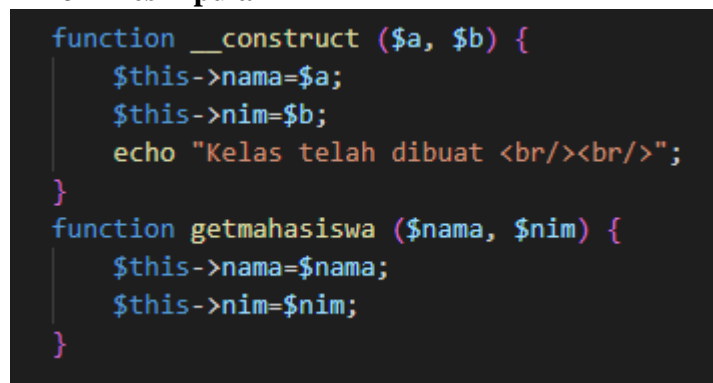
- **Latihan 2**

- Tampilan dari sebelum dan sesudah dimodifikasi



```
localhost/tugas/Lat4_2b.php  
Kelas telah dibuat  
  
Pennyka  
0910683073  
  
Kelas telah dihancurkan
```

- **Kesimpulan**

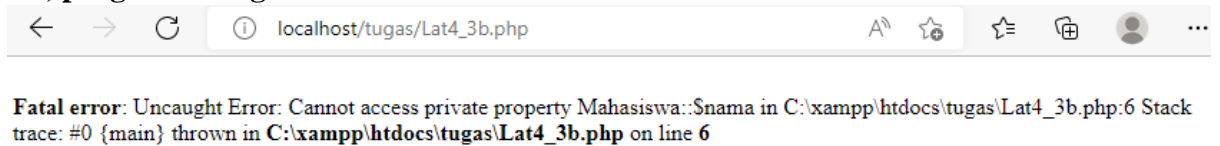


```
function __construct ($a, $b) {  
    $this->nama=$a;  
    $this->nim=$b;  
    echo "Kelas telah dibuat <br/><br/>";  
}  
function getmahasiswa ($nama, $nim) {  
    $this->nama=$nama;  
    $this->nim=$nim;  
}
```

Dengan menggunakan konstruktor seperti di atas akan memudahkan dalam memasukkan suatu data jika data yang akan dimasukkan sangat banyak. Jadi kita tidak perlu menulis satu-persatu.

- **Latihan 3**

- a. **Ya, program mengalami ke-erroran.**



Karena file yang akan diakses terprivate.

- b. **Protected**

Fatal error: Uncaught Error: Cannot access protected property Mahasiswa::\$nama in C:\xampp\htdocs\tugas\Lat4_3b.php:6 Stack trace: #0 {main} thrown in C:\xampp\htdocs\tugas\Lat4_3b.php on line 6

Public

hendra

- c. **Memodifikasi Lat4_3b.php sehingga bisa mencetak isi dari nim dan nama dengan modifier private**

Nama: hendra
NIM: 001253649

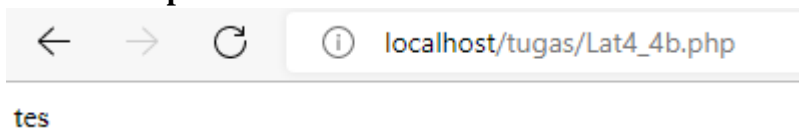
Sealamat Anda terdaftar sebagai member

- d. **Kesimpulan:**

Dengan adanya modifier private dan protected maka data tidak akan bisa diakses karena terlindungi maka diperlukan fungsi “**echo ("Nama: " . \$mhs1->getNama());**” agar data bisa diakses. Tetapi, apabila modifier sudah dalam bentuk public maka bisa diakses tanpa tambahan fungsi seperti di atas, hanya perlu memasukkan “**echo \$mhs1->nama;**” maka data akan muncul.

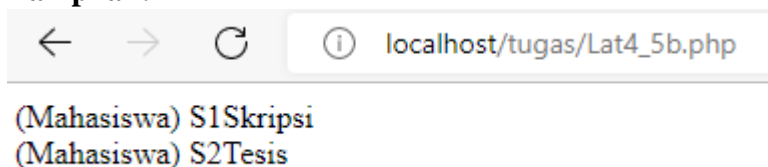
- **Latihan 4**

- o **Tampilan:**



- **Latihan 5: Abstract Class**

- a. **Tampilan:**



b. Tampilan setelah kode baris 29-32 pada Lat4_5a.php dihapus.

Fatal error: Class Magister contains 1 abstract method and must therefore be declared abstract or implement the remaining methods (Mahasiswa::getTugasAkhir) in C:\xampp\htdocs\tugas\Lat4_5a.php on line 26

Karena pada class Magister terdapat satu abstract method yang mana method tersebut harus diterapkan; (*Mahasiswa::getTugasAkhir*).

c. Kesimpulan:

Dari latihan 5 tersebut dapat disimpulkan apabila kita tidak mengaplikasikan salah satu method yang ada maka akan terjadi kesalahan karena abstract method harus diterapkan.

• **Latihan 6**

a. Maksud dari program yang ada di Lat4_6.php

Dalam objek interface memudahkan kita untuk membuat kode yang menentukan method mana yang akan diimplementasikan tanpa harus mendefinisikan bagaimana method tersebut bekerja.

Class sendiri tidak bisa menerapkan dua interface yang mempunyai nama method yang sama. Interface dapat mewariskan seperti class dengan menggunakan extends. Class yang menerapkan interface harus menggunakan method yang ada pada interface tersebut.

b. Tampilan ketika kode baris 27-29 dihapus

Fatal error: Class d contains 1 abstract method and must therefore be declared abstract or implement the remaining methods (c::baz) in C:\xampp\htdocs\tugas\Lat4_6.php on line 17

Tentu saja mengalami *error* karena ada satu method yang tidak diterapkan, karena pada *interface c extends a, b* terdapat method *public function baz ()*, dan class d terimplements dengan interface c yang mana sebuah class harus menerapkan semua method yang ada pada interface yang diimplementasikan.

c. Class e yang mempunyai method *foo* dan *bar*

```
interface m extends a
{
    public function bar ();
}

class e implements m
{
    public function foo ()
    {
    }

    public function bar ()
    {
    }
}

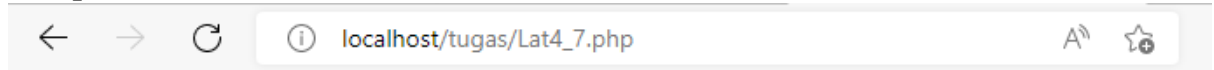
?>
```

d. Kesimpulan:

Jika suatu class mengimplementasikan suatu interface maka class tersebut harus menggunakan method yang digunakan pada interface tersebut. Jika ada satu saja method yang tidak diterapkan maka akan terjadi kesalahan.

• **Latihan 7**

a. Tampilan:



Fatal error: Cannot override final method A::disp() in C:\xampp\htdocs\tugas\Lat4_7.php on line 16

Karena kata “final” digunakan untuk mencegah proses overriding method sehingga jika meletakkan kata final pada method maka method akan berhenti dan tidak bisa diperpanjang.

b. Tampilan setelah memodifikasi dengan menghapus kata final pada baris 5 dan menambahkan kata final pada baris 2.

Fatal error: Class B may not inherit from final class (A) in C:\xampp\htdocs\tugas\Lat4_7.php on line 22

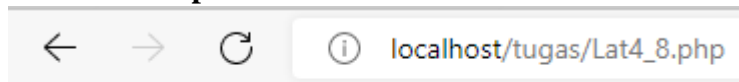
Karena ketika menggunakan kata “final” maka class tidak dapat diperpanjang atau diwariskan pada anak class. Dan terjadilah **fatal error**.

c. Kesimpulan:

Jika kita menggunakan kata “final” maka tidak akan bisa dioverride, ini berlaku pada method dan class, karena sifat dari kata “final” adalah mencegah proses overriding method pada anak class, sehingga suatu class yang menggunakan kata “final” tidak akan bisa diwariskan,

• **Latihan 8**

○ **Kesimpulan:**



0

Karena pada method terdapat fungsi *print self::\$var* yang mana hal itu akan mencetak sesuai dengan method yang dituliskan dibawah class; *private static \$var=0*

• **Latihan 9**

1. Model yang berhubungan langsung dengan database mysql.
2. View sebuah object dengan susunan class view yang mengambil file-file template.

JUDUL	PENGARANG	PENERBIT	TAHUN
Pembunuhan di Orient Express	Agatha Christie	Collins Crime Club	1934
Pembunuhan di Sungai Nil	Agatha Christie	Collins Crime Club	1937
Pembunuhan ABC	Agatha Christie	Collins Crime Club	1936
Misteri di Styles	Agatha Christie	Collins Crime Club	1920